

Git & github

Hadley Wickham

@hadleywickham

Chief Scientist, RStudio



February 2015

Overview

- Focus mostly on the why, not the how
- Learning git & github is definitely frustrating at first
- Two big pay offs: **safety & community**
- Mechanics at <http://r-pkgs.had.co.nz/git.html>

Safety


“Failures, repeated failures,
are finger posts on the road
to achievement. One fails
forward toward success.”


















— C. S. Lewis





A STORY TOLD IN FILE NAMES:

Location:  C:\user\research\data

Filename	Date Modified	Size	Type
 data_2010.05.28_test.dat	3:37 PM 5/28/2010	420 KB	DAT file
 data_2010.05.28_re-test.dat	4:29 PM 5/28/2010	421 KB	DAT file
 data_2010.05.28_re-re-test.dat	5:43 PM 5/28/2010	420 KB	DAT file
 data_2010.05.28_calibrate.dat	7:17 PM 5/28/2010	1,256 KB	DAT file
 data_2010.05.28_huh??.dat	7:20 PM 5/28/2010	30 KB	DAT file
 data_2010.05.28_WTF.dat	9:58 PM 5/28/2010	30 KB	DAT file
 data_2010.05.29_aaarrgh.dat	12:37 AM 5/29/2010	30 KB	DAT file
 data_2010.05.29_#\$_*!&.dat	2:40 AM 5/29/2010	0 KB	DAT file
 data_2010.05.29_crap.dat	3:22 AM 5/29/2010	437 KB	DAT file
 data_2010.05.29_notbad.dat	4:16 AM 5/29/2010	670 KB	DAT file
 data_2010.05.29_woohoo!.dat	4:47 AM 5/29/2010	1,349 KB	DAT file
 data_2010.05.29_USETHISONE.dat	5:08 AM 5/29/2010	2,894 KB	DAT file
 analysis_graphs.xls	7:13 AM 5/29/2010	455 KB	XLS file
 ThesisOutline!.doc	7:26 AM 5/29/2010	38 KB	DOC file
 Notes_Meeting_with_ProfSmith.txt	11:38 AM 5/29/2010	1,673 KB	TXT file
 JUNK...	2:45 PM 5/29/2010		Folder
 data_2010.05.30_startingover.dat	8:37 AM 5/30/2010	420 KB	DAT file

Type: Ph.D Thesis Modified: too many times Copyright: Jorge Cham www.phdcomics.com

"FINAL".doc



FINAL.doc!



FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc

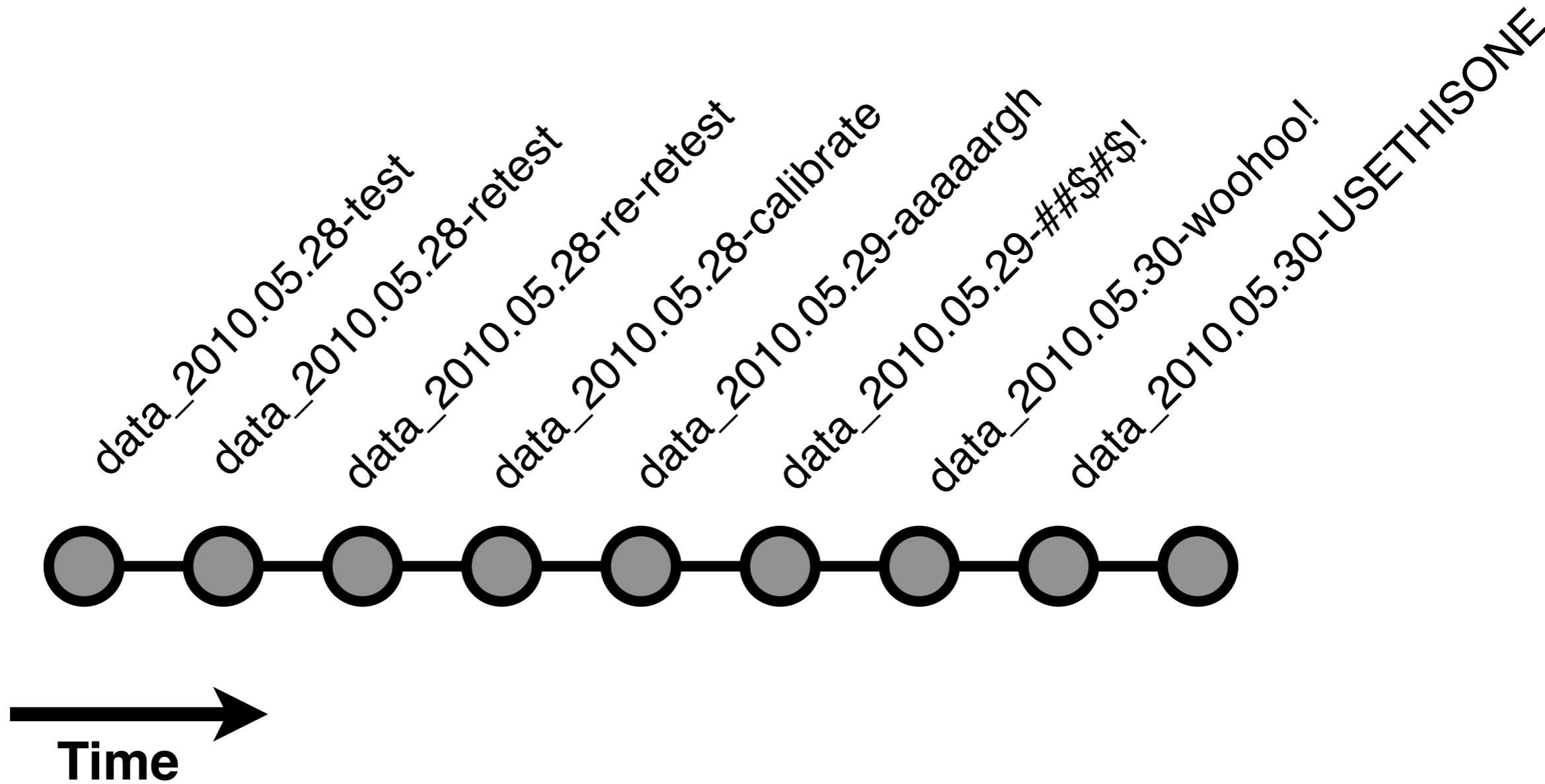


FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSCHOOL?????.doc





The history of a project can be viewed as a series of changes

Changes

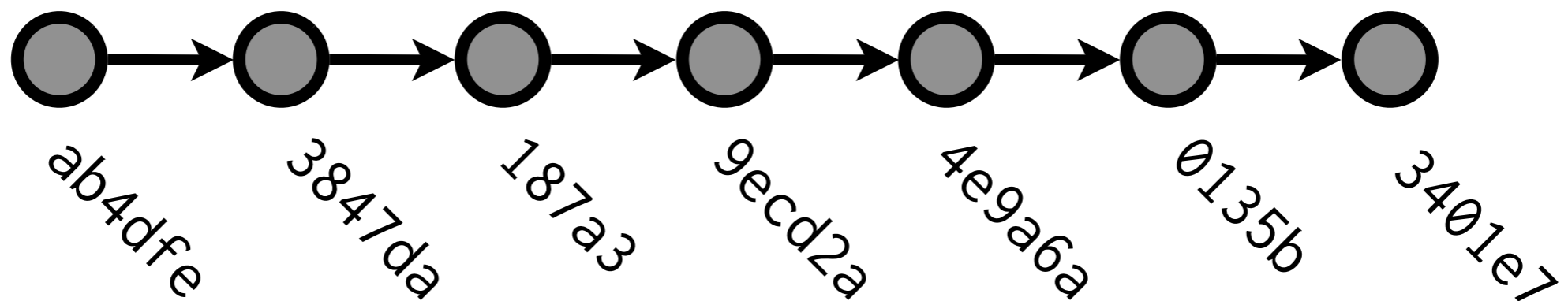
- A unique identifier
- What changed?
- When did it change?
- Who changed it?
- Why did it change?

Changes

- **A unique identifier**
- What changed?
- **When did it change?**
- Who changed it?
- **Why did it change?**

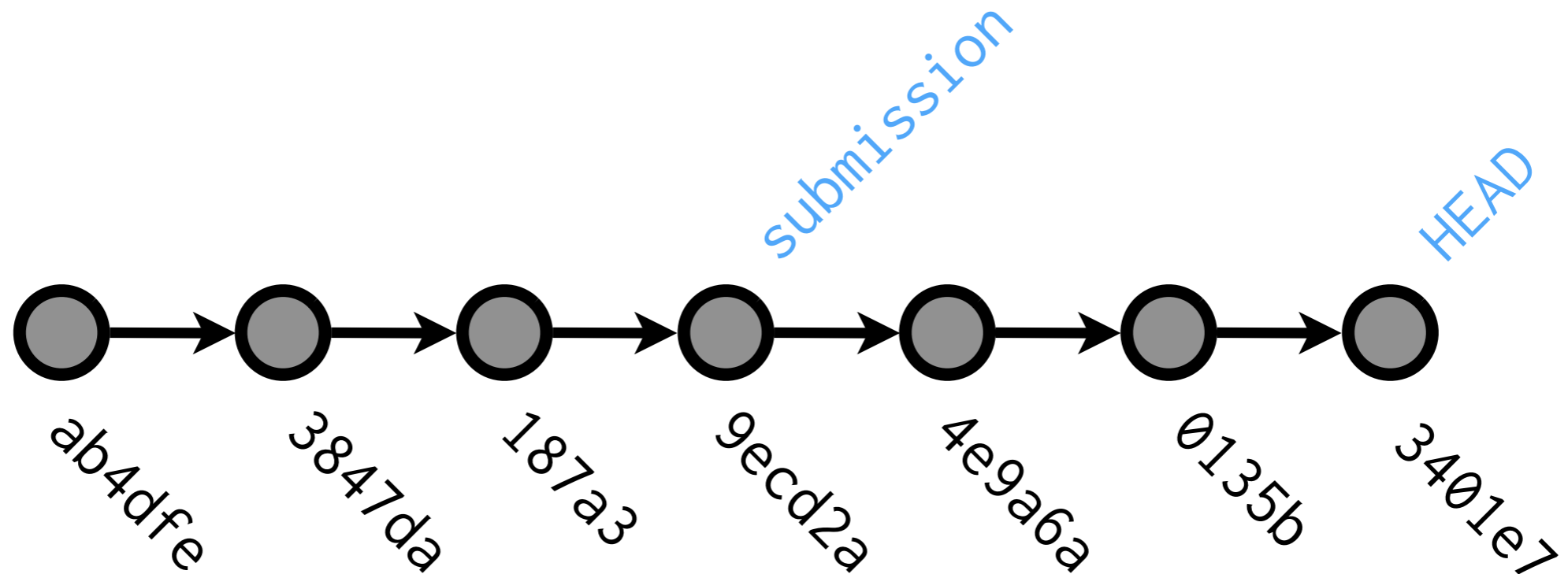
(And this obviously gets hard to coordinate for multiple files)

With git, each change (**commit**) is given a unique identifier, called a **sha**



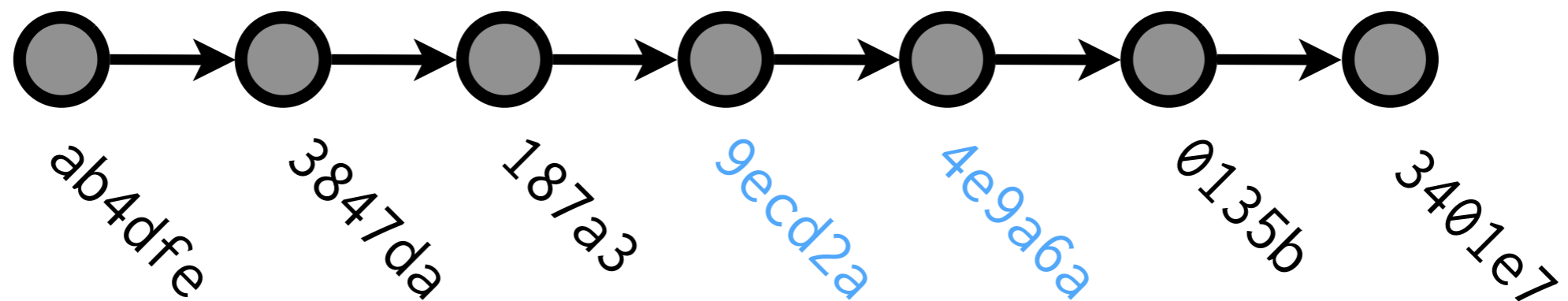
The sha is a key into a database that provides the author, date, and a description

You can also name individual commits



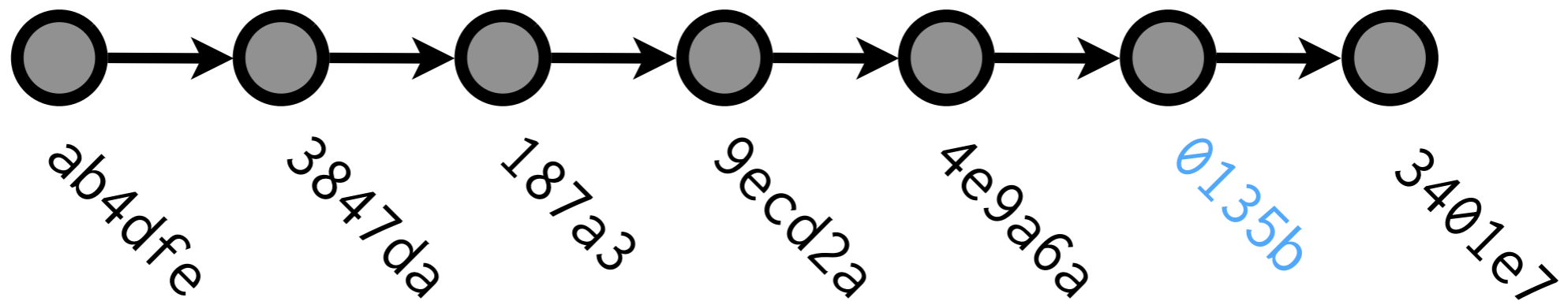
```
git tag submission 9ecd2a
```

Then see exactly what's changed



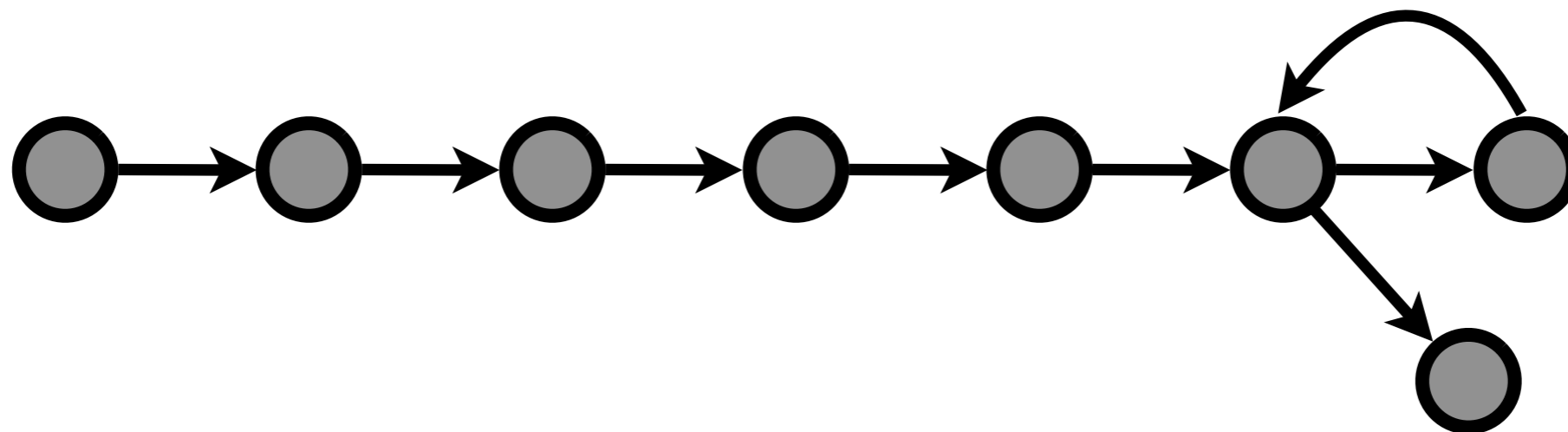
```
git diff 9ecd2a..4e9a6a
```

You can revert to a previous change with git checkout

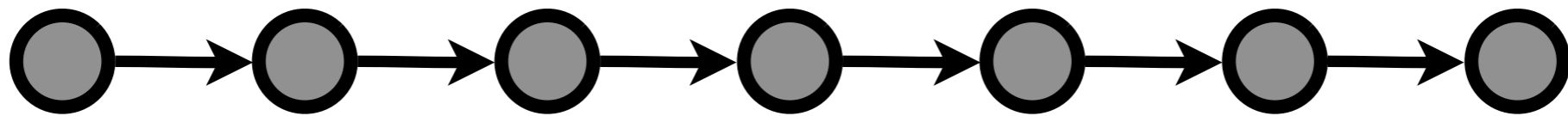


```
git checkout 0135b
```

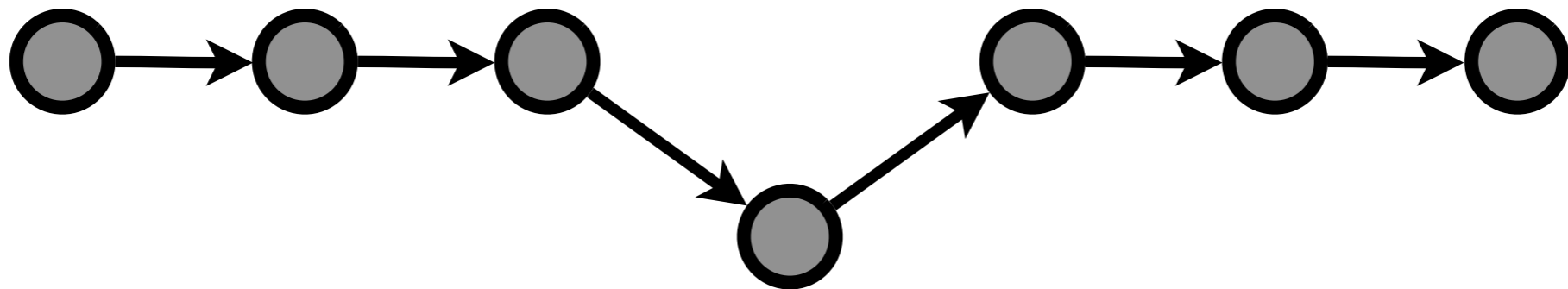
That allows you to undo mistakes



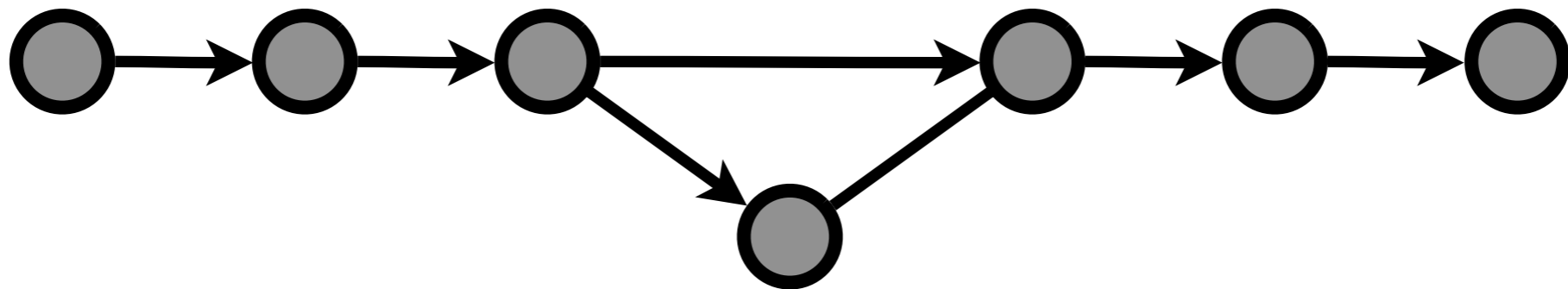
Alternatively you can go back in time and pretend mistakes never existed



Alternatively you can go back in time and pretend mistakes never existed



Alternatively you can go back in time and pretend mistakes never existed. **That's risky!**



`git rebase ...`

Demo

<http://r-pkgs.had.co.nz>

<http://github.com/hadley/r-pkgs>

Demo

Vows package & roxygen2

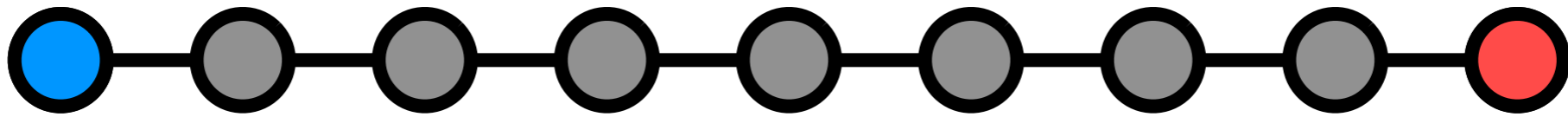
Other benefits

git bisect

<https://github.com/wch/bisectr>

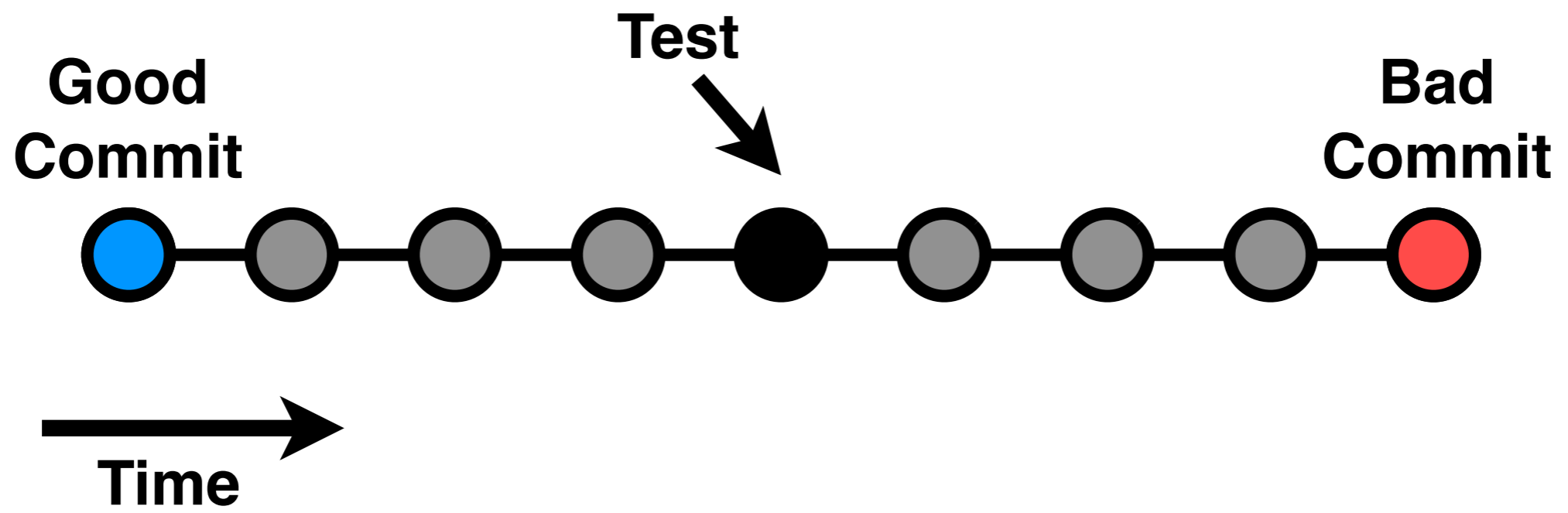
**Good
Commit**

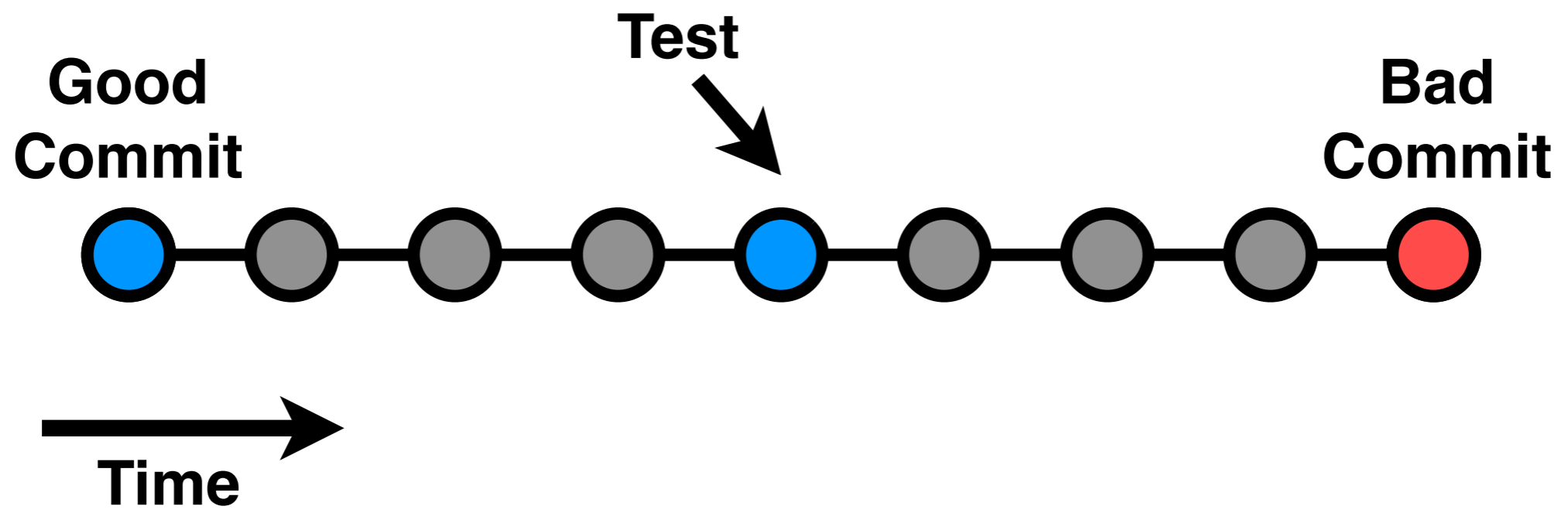
**Bad
Commit**



Time

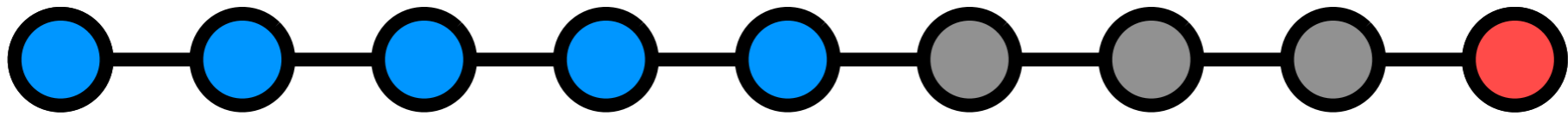
**Where did the
problem
happen?**

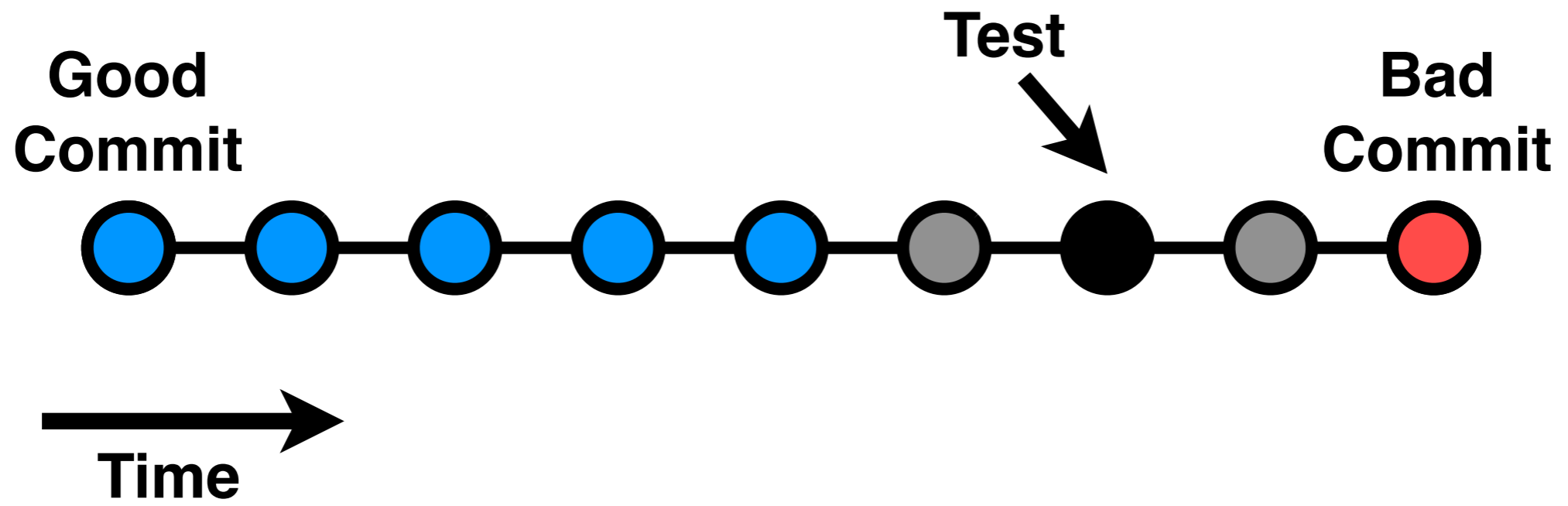


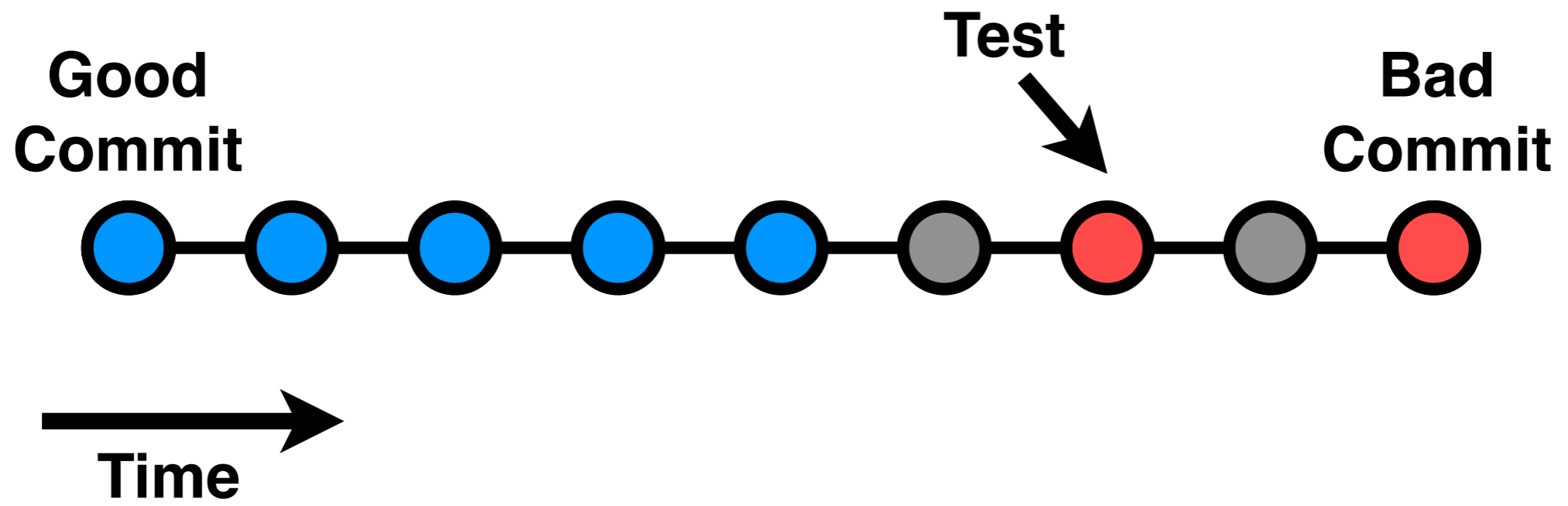


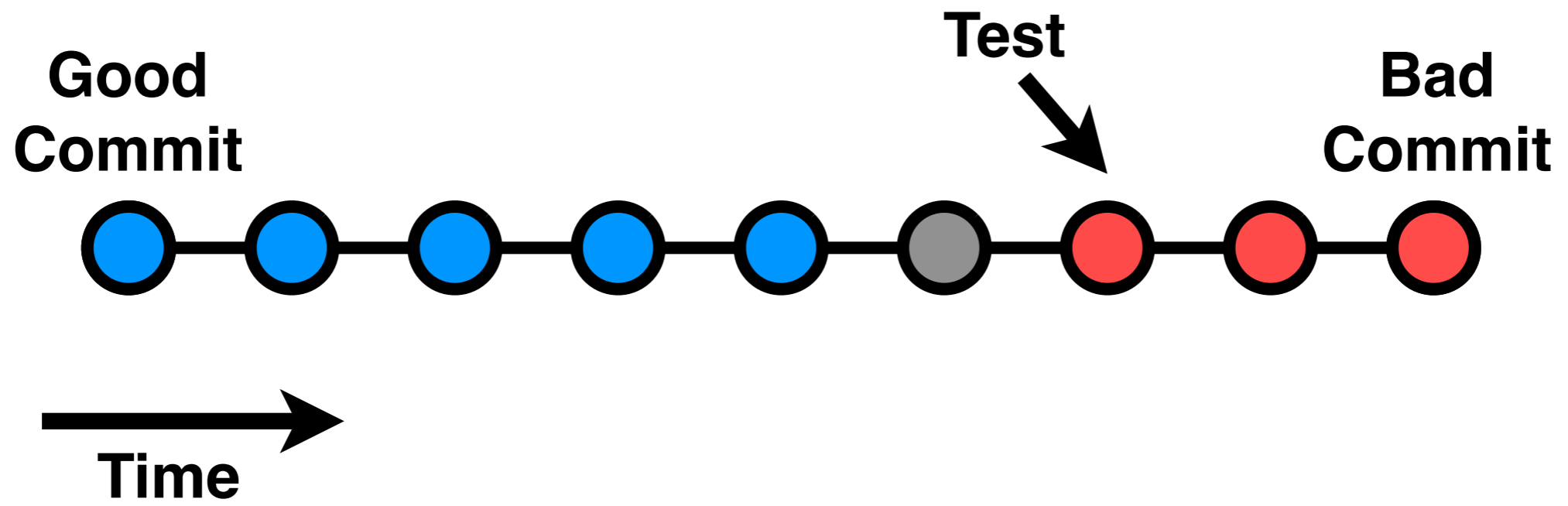
**Good
Commit**

**Bad
Commit**









Community

A home on the web

The screenshot shows the GitHub repository page for 'hadley/dplyr'. The browser address bar shows 'https://github.com/hadley/dplyr'. The repository name 'hadley / dplyr' is displayed at the top left. The repository statistics are: 935 commits, 7 branches, 0 releases, and 4 contributors. The repository description is 'Plyr specialised for data frames: faster & with remote datastores'. The right sidebar contains navigation links: Code, Issues (25), Pull Requests (0), Wiki, Pulse, Graphs, Network, and Settings. The SSH clone URL is 'git@github.com:hadley:dplyr'. The main content area shows a list of files and folders with their commit messages and dates.

GitHub, Inc. [US] <https://github.com/hadley/dplyr>

This repository Search or type a command Explore Gist Blog Help hadley

hadley / dplyr Unwatch 31 Star 109 Fork 19

Plyr specialised for data frames: faster & with remote datastores — Edit

935 commits 7 branches 0 releases 4 contributors

branch: master dplyr / Show language statistics

more on #152

romainfrancois authored 2 days ago latest commit a91dc1206d

R	Only create if <code>_doesn't_ exist</code>	3 days ago
data	Recompress nasa data	3 days ago
demo	Add trailing nl	13 days ago
inst	more on #152	2 days ago
man-roxygen	Shift manip docs into individual srcs	2 months ago
man	Fix incorrect uses of <code>ll</code> with <code>l</code>	3 days ago
src	more on #152	2 days ago
tests	Move to cran version of testthat	3 days ago
vignettes	Vignettes can't use <code>error = TRUE</code>	4 days ago
.Rbuildignore	Ignore man-roxygen in built package	a month ago
.gitignore	Get vignettes workign	2 months ago

SSH clone URL
git@github.com:hadley:dplyr

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Clone in Desktop Download ZIP

Nicely formatted readme

README.md

dplyr

build failing

dplyr is the next iteration of plyr, focussed on tools for working with data frames (hence the `d` in the name). It has three main goals:

- Identify the most important data manipulation tools needed for data analysis and make them easy to use from R.
- Provide blazing fast performance for in-memory data by writing key pieces in C++.
- Use the same interface to work with data no matter where it's stored, whether in a data frame, a data table or database.

`dplyr` is not yet available on CRAN, but you can install it from github with:

```
devtools::install_github("dplyr")
```

To get started, read the notes below, then read the intro vignette: `vignette("introduction", package = "dplyr")`. To make the most of dplyr, I also recommend that you familiarise yourself with the principles of [tidy data](#): this will help you get your data into a form that works well with dplyr, ggplot2 and R's many modelling functions.

If you encounter a clear bug, please file a minimal reproducible example on [github](#). For questions and other discussion, please use the [manipulatr mailing list](#).


```
# Easy to install a package from github
library(devtools)
install_github("hadley/devtools")
```

```
# Github is a viable location for package
# distribution, but it doesn't yet have the
# reach or authority of CRAN. Currently
# about ~3,000 packages on github.
```

See what's happening

The screenshot shows the GitHub interface for the 'dplyr' repository. The page title is 'hadley / dplyr' and the current branch is 'master'. The commit history is displayed in reverse chronological order, with the most recent commit at the top. Each commit entry includes a profile picture of the author (romainfrancois), the commit message, the time since it was authored, and a 'Browse code' link with the commit hash.

Date	Commit Message	Author	Time Ago	Commit Hash	Action
Dec 07, 2013	more on #152	romainfrancois	2 days ago	a91dc1206d	Browse code →
Dec 06, 2013	using tbl_df instead of raw data.frame for the test	romainfrancois	2 days ago	864c8b7b9f	Browse code →
	fix test	romainfrancois	3 days ago	58eaddc3b3	Browse code →
	using NamedListAccumulator	romainfrancois	3 days ago	0c7d40db1a	Browse code →
	handling date and time constants	romainfrancois	3 days ago	457dd7e8f6	Browse code →
	handling special cases in summarise. closes #153	romainfrancois	3 days ago	eb2c3566f4	Browse code →
	summarise handles constants (#153)	romainfrancois	3 days ago	ea873897a9	Browse code →
	dealing with the case summarise(df, log(z)) from #153	romainfrancois	3 days ago	7d0fe6d829	Browse code →
	dealing with the case summarise(df, diff(z)) from #153	romainfrancois	3 days ago	d278dfcec8	Browse code →

See exactly what changed

GitHub, Inc. [US] <https://github.com/hadley/dplyr/commit/864c8b7b9fae5329d99d818bd78b2b1686bdb167>

This repository Search or type a command Explore Gist Blog Help hadley

hadley / dplyr Unwatch 31 Star 109 Fork 19

using `tbl_df` instead of raw `data.frame` for the test

master

romainfrancois authored 2 days ago 1 parent [58eaddc](#) commit [864c8b7b9fae5329d99d818bd78b2b1686bdb167](#)

Showing 2 changed files with 3 additions and 3 deletions. Show Diff Stats

2 files changed

inst/tests/test-summarise.r View file @ 864c8b7

```
@@ -69,7 +69,7 @@ test_that("summarise handles constants (#153)", {
69 69   today <- Sys.Date()
70 70   now <- Sys.time()
71 71
72 - res <- summarise(df, int = 1L, num = 1.0,
72 + res <- summarise(tbl_df(df), int = 1L, num = 1.0,
73 73   str = "foo", bool = TRUE, date = today, time = now)
74 74   expect_equal(res$int, 1L)
75 75   expect_equal(res$num, 1.0)
```

src/dplyr.cpp View file @ 864c8b7

```
@@ -1024,14 +1024,14 @@ SEXP summarise_not_grouped(DataFrame df, List args, Environment env){
1024 1024   Rcpp::Shelter<SEXP> __ ;
1025 1025   for( int i=0; i<nexpr; i++){
1026 1026     SEXP name = names[i] ;
1027 -
1028 -     boost::scoped_ptr<Result> res( get_handler( args[i], subsets, env ) ) ;
1027 +     Result* res = get_handler( args[i], subsets, env ) ;
```

Track issues

The screenshot shows the GitHub interface for the 'hadley/dplyr' repository. The browser address bar shows the URL: <https://github.com/hadley/dplyr/issues?milestone=2&state=open>. The repository name 'hadley / dplyr' is displayed at the top left, with statistics for 'Unwatch' (31), 'Star' (109), and 'Fork' (19) on the right. The main content area shows a list of 4 open issues, sorted by 'Newest'. The issues are:

- .data.frame versions not using the correct environment** (#159), opened by romainfrancois 2 days ago, 2 comments.
- Mutate fails when creating a constant** (#152), opened by hadley 4 days ago, 8 comments, labeled 'bug'.
- C++ code needs to regularly check for user interrupts** (#123), opened by hadley 21 days ago, 4 comments, labeled 'enhancement'.
- Plyr compatibility** (#29), opened by hadley 4 months ago, 2 comments, labeled 'enhancement'.

On the left sidebar, there are filters for 'Everyone's Issues' (4), 'Assigned to you' (0), 'Created by you' (3), and 'Mentioning you' (1). Below these is a 'Milestone: v0.1' section with a progress bar and '4 open issues · Due in 7 days'. The 'Labels' section shows 'bug' (1) and 'enhancement' (2). A 'New label' input field is at the bottom left. The right sidebar contains navigation icons for code, info, issues, calendar, activity, charts, and settings.

Talk things over

The screenshot shows a GitHub issue page for the repository 'hadley/dplyr'. The issue title is 'C++ code needs to regularly check for user interrupts', opened by 'hadley' 21 days ago. It is assigned to 'romainfrancois' and is part of the 'v0.1' milestone. The issue is labeled as 'enhancement' and is currently 'Open'. There are 4 comments. The first comment, from 'romainfrancois' 14 days ago, references a commit 'romainfran... adding code to check for interrupts. #123'. A second comment from 'romainfrancois' 14 days ago discusses adding calls to 'R_CheckUserInterrupt' and asks for feedback on C++ compatibility. A third comment from 'romainfrancois' 14 days ago references a thread on R-devel and includes a code snippet for a C++ interrupt check function.

hadley opened this issue 21 days ago

C++ code needs to regularly check for user interrupts

romainfrancois is assigned Milestone: v0.1

So if you accidentally miss-specify a call you can abort it.

2 participants

romainfrancois referenced this issue from a commit 14 days ago

romainfran... adding code to check for interrupts. #123

romainfrancois commented 14 days ago

I'm adding calls to `R_CheckUserInterrupt` here and there. But I'm wondering if this plays well with C++.

I think this uses long jumps and that possibly bypasses C++ destructors. I might ask this on R-devel.

romainfrancois commented 14 days ago

This comes from Simon's contribution to [a thread on R-devel](#) :

```
static void chkIntFn(void *dummy) {
  R_CheckUserInterrupt();
}

// this will call the above in a top-level context so it won't longjmp-out of your context
bool checkInterrupt() {
  return (R_ToplevelExec(chkIntFn, NULL) == FALSE);
}

// your code somewhere ...
```

Receive pull requests

The screenshot shows a GitHub pull request page for the repository 'hadley / dplyr'. The page title is 'hadley / dplyr' and the pull request number is '#51'. The pull request is titled 'fix bug: mysql connection password was not passed' and was opened by 'szilard' 2 months ago. It has been merged into 'hadley:master' from 'szilard:master' about 1 month ago. The pull request is marked as 'Merged' and has 2 commits. The description of the pull request is 'it was working (only) if user had no password'. The pull request is associated with a commit '3ebb93f' by 'szilard' and a merge commit 'ebc7672' by 'hadley'. The pull request is also marked as 'Closed' by 'hadley' 2 months ago. The page shows the repository name 'hadley / dplyr', the number of stars (109), forks (19), and unwatchers (31). The page also shows the repository's navigation menu (Explore, Gist, Blog, Help) and the user's profile (hadley).

hadley / dplyr Unwatch 31 Star 109 Fork 19

Merged hadley merged 1 commit into hadley:master from szilard:master about 1 month ago 2 #51

Conversation Commits 1 Files Changed 1

szilard opened this pull request 2 months ago
fix bug: mysql connection password was not passed
Edit

No one is assigned No milestone

it was working (only) if user had no password

szilard added a commit 2 months ago
szilard fix bug: mysql connection password was not passed 3ebb93f

hadley referenced this pull request from a commit 2 months ago
hadley Merge pull request #51 from szilard/master ebc7672

Merged hadley merged commit ebc7672 into hadley:master from szilard:master 2 months ago

Closed hadley closed the pull request 2 months ago

... and review online

GitHub, Inc. [US] <https://github.com/hadley/dplyr/pull/51/files>

This repository Search or type a command Explore Gist Blog Help hadley

hadley / dplyr Unwatch 31 Star 109 Fork 19

Merged hadley merged 1 commit into hadley:master from szilard:master about 1 month ago 2 #51

Conversation Commits 1 Files Changed 1

Showing 1 changed file with 1 addition and 1 deletion. Show Diff Stats

R/src-mysql.r Open View file @ 3ebb93f

```
@@ -95,7 +95,7 @@ src_mysql <- function(dbname, host = NULL, port = 0L, user = "root",
95 95     }
96 96
97 97     con <- dbi_connect(MySQL(), dbname = dbname , host = host, port = port,
98 -   username = user, ...)
98 +   username = user, password = password, ...)
99 99     info <- db_info(con)
100 100
101 101     src_sql("mysql", con,
```

Tip: You can add notes to lines in a file. Hover to the left of a line to make a note

© 2013 GitHub, Inc. Terms Privacy Security Contact Status API Training Shop Blog About

Find out as soon as you break something

The screenshot shows the Travis CI web interface. The top navigation bar includes the Travis logo, links for Home, Blog, Status, Help, and Travis CI for Private Repositories. A search bar is located on the left. Below the search bar are tabs for 'My Repositories' and 'Recent'. A list of repositories is shown, with 'klutometis/roxygen' highlighted in green, indicating a passing build. The right side of the interface shows the details for the 'klutometis/roxygen' repository, including a 'build passing' status and a 'Current' build tab. The build details show a successful build (97) for commit b510eaa (master) by author hadley. The build message is 'First pass at functions in usage. #164'. Below the build details is a terminal window showing the build script execution, including cloning the repository, checking out the commit, and running tests.

Travis CI interface showing repository status and build details for `klutometis/roxygen`.

Build Status: **build passing**

Build Details:

- Build: **97**
- State: Passed
- Finished: about 2 hours ago
- Duration: 3 min 55 sec
- Message: First pass at functions in usage. #164
- Commit: [b510eaa \(master\)](#)
- Compare: [93fe512416d8...b510eaab00df](#)
- Author: [hadley](#)
- Committer: [hadley](#)

Terminal Output:

```
1 Using worker: worker-linux-5-2.bb.travis-ci.org:travis-linux-3
2
3 $ export global=[
4 $ export WARNINGS_ARE_ERRORS=1
5 $ export CC=gcc
6 $ git clone --depth=50 --branch=master git.1
13 $ cd klutometis/roxygen
14 $ git checkout -qf b510eaab00df548d02305d5f233dab3d54420b9f git.3
15 $ gcc --version
16 gcc (Ubuntu/Linaro 4.6.3-1ubuntu5) 4.6.3
17 Copyright (C) 2011 Free Software Foundation, Inc.
18 This is free software; see the source for copying conditions. There is NO
19 warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
20
21 $ curl -OL http://raw.githubusercontent.com/craigcitro/r- before_install.1
26 $ chmod 755 ./travis-tool.sh before_install.2
27 $ ./travis-tool.sh bootstrap before_install.3
322 $ ./travis-tool.sh install_deps install
574 $ ./travis-tool.sh run_tests
575 + CRAN=http://cran.rstudio.com
```


Beautiful release notes

The screenshot shows the GitHub interface for the repository 'klutometis/roxygen'. The page is titled 'Releases / Tags' and features a 'Draft a new release' button. The latest release is 'roxygen2 3.0.0', which was released 3 days ago by 'hadley' and includes 3 commits. The release notes describe the new features and improvements in this version, such as full support for S4 and RC (reference classes), better default usage definitions, and improved documentation for S4 classes and methods.

Latest release

v3.0.0
abe5d0d

roxygen2 3.0.0

hadley released this 3 days ago · 3 commits to master since this release

Roxygen2 now fully supports S4 and RC (reference classes) - you should no longer need to manually add `@alias` or `@usage` tags for S4 classes, methods and generics, or for RC classes.

- The default usage definitions are much better, generating the correct usage for data sets ([#122](#)), S3 methods (without additional `@method` tag), S4 generics, S4 methods, and for replacement ([#119](#)) and infix functions. Backslashes in function arguments in are correctly escaped. Usage statements also use a more sophisticated line wrapping algorithm so that they should cause fewer problems with the R CMD check line limit. ([#89](#), [#125](#)).
- S4 classes, S4 methods, and RC classes are given better default topics, and the file names corresponding to those topics are shorter.
- S4 methods will automatically inherit parameter documentation from their generic.
- `@slot name description` allows you to document the slots of a S4 class.

Demo

Easily contribute small fixes

<https://github.com/mjockers/syuzhet>

My advice

Learning Git & Github

- Is frustrating at first & there's a lot of new vocab to learn.
- Pays off by increasing the pace of your coding because you don't have to worry as much about mistakes.
- Pays off by connecting you with a community of like minded people who can help make your code better.

<http://r-pkgs.had.co.nz/git.html>